Leaflet Maps for Contao Dokumentation

Release 3.0.0

netzmacht David Molineus

17.04.2019

Inhaltsverzeichnis

1 Benutzerhandbuch **1**
- 1.1 Benutzerhandbuch . . . 1
  - 1.1.1 Installation . . . 1
  - 1.1.2 Erste Schritte . . . 2
  - 1.1.3 Konzepte . . . 2
  - 1.1.4 Karten & Layer verwalten . . . 3
  - 1.1.5 Karten auf der Website einbinden . . . 5
  - 1.1.6 Anpassen . . . 5
  - 1.1.7 Erweiterungen . . . 6

2 Entwicklerhandbuch **11**
- 2.1 Entwicklerhandbuch . . . 11

3 Lizenz **13**

4 Mitwirkende **15**

5 Verzeichnisse **17**

KAPITEL 1

Benutzerhandbuch

1.1 Benutzerhandbuch

1.1.1 Installation

Systemvoraussetzungen

Leaflet für Contao in der Version 3 bringt folgende Systemvoraussetzungen mit:

- min. Contao 4.4
- min. PHP 7.1

Über den Contao Manager

Leaflet für Contao lässt sich komfortabel über den Contao Manager installieren:

1. netzmacht/contao-leaflet-maps suchen
2. Gewünschte Version eingeben (^3.0.0) auswählen
3. Änderungen anwenden
4. Install-Tool aufrufen und die Datenbank aktualisieren
5. Fertig!

Manuell mit Composer

Desweiteren kann die Erweiterung manuell über Composer installiert werden.

1. php composer require netzmacht/contao-leaflet-maps:^3.0.0
2. Install-Tool aufrufen und die Datenbank aktualisieren
3. Fertig!

1.1.2 Erste Schritte

Nach der erfolgreichen Installation von Leaflet für Contao erscheint im Contao Backend in der linken Navigationssidebar ein zusätzlicher Bereich *Leaflet*.

Um die erste Karte zu konfigurieren sind folgende Schritte zu empfehlen:

1. Kachel-Layer definieren

Bevor die erste Karte definiert wird, ist es empfehlenswert die Layer zu definieren, da Layer separat von den Karten angelegt und in der Karte nur referenziert werden.

Die Kachel-Layer generieren die Karte selbst. Wie Leaflet ist der Anwender hier vollkommen frei eigene Kartenanbieter zu integrieren. Um nicht für jeden Anbieter eine manuelle Konfiguration vorzunehmen, unterstützt die Erweiterung mit dem Layertyp **Vorkonfigurierte Karte** alle Karten die von dem Plugin **'Leaflet Providers'_** angeboten werden.

Als Karten-Layer legen wir daher nun ein Layer vom Typ **Vorkonfigurierte Karte** aus und wählen als **Kartenanbieter** *OpenStreetMap* aus und aktivieren das Layer.

2. Marker-Layer definieren

Alle Daten werden bei Leaflet für Contao unter Layern verwaltet. Bevor die Marker selbst angelegt werden können, muss deshalb ein Layer vom Typ **Marker** angelegt werden. Neben der Angabe des Titels ist es erfoderlich den Layer unter *Aktivierung* zu aktivieren.

Schließt man nun die Bearbeitungsmaske erscheint in der Layeransicht ein Bleistift. In der von Contao gewohnen Struktur (z.B. Inhaltselemente von Artikeln) können nun die Marker angelegt werden. Einzugeben sind die Werte für *Title* und *Koordinaten*. Wie bereits bei den Layern ist außerdem das Aktivieren erforderlich.

Statt der manuellen Eingabe der Koordinaten (Kommasepararierte Werte für *latitute,longitude[,altidute]*), kann auf das Karten-Widget zur Geokodierung einer eingebenen Adresse verwendet werden.

3. Karte anlegen

Nun ist alles vorbereitet um die erste Karte zu erstellen. Im Modul **Leaflet-Karten** kann nun eine neue Karte angelegt werden. [TODO]

1.1.3 Konzepte

Trennung von Karten und Layern

Leaflet für Contao trennt die Datenverwaltung von der Darstellung. Sämtliche Kartenelemente werden unter Layer verwaltet. Ob Tile-Layer, Marker, Vektoren, Markercluster, ... - alles wird unter Layer angelegt. Karten dienen dazu die verschiedenen Layeer miteinander zu verbinden und das Verhalten der Karte zu bestimmen.

Generierung von Javascript

Leaflet für Contao generiert aus der Datebankdefinition automatisch den entsprechenden Javascript-Code. Dieser wird, anders als bei anderen Erweiterungen nicht im Template zusammengesetzt. Das Verhalten der Generierung lässt sich demnach kaum über das Template steuern. Vielmehr beitet Leaflet für Contao einige Schnittstellen auf Programmierebene, wo die Ausgabe manipuliert werden kann.

Dynamisches Laden

Aus den Kartendefinitionen wird Javascript generiert, welcher in das Markup der Seite geladen wird. Bei großen Datenmengen würde dies unnötig den Quelltext aufblähen und die Ladezeiten negativ beeinflussen. Daher können Daten auch dynamisch geladen werden. Dynamsich geladene Daten können z.b. anhand der Kartengrenzen beegrenzt werden.

Datenformat GeoJSON

Leaflet für Contao nutzt als zentrales Datenformat GeoJSON. Selbst wenn im Backend Marker definiert werden, werden diese als GeoJSON Daten übermittelt und erst On The Fly in Marker generiert. Dies ermöglicht zum einem das dynamische Nachladen der Daten via AJAX als auch die individuelle Konvertierung der GeoJSON Features in die Kartenfeatures.

Kartengrenzen

Ergänzend zu den Standadfunktionen von Leaflet bietet die Erweiterung die Möglichkeit die Grenzen einer Karte unter verschiedenen Bedingungen zu verändern:

- statisches Festlegen der Grenzen anhand der initialen Konfiguration
- Laden von Daten innerhalb der Kartengrenzen
- Daten erweitern die Kartengrenzen initial
- Daten erweitern die Kartengrenzen, auch wenn diese dynamisch geladen werden

Das grundlegende Verhalten ist bei der Karte zu konfigurieren. Bei den einzelnen Layern kann dann das individuelle Verhalten der Layer-Daten definiert werden.

Caching

Leaflet für Contao generiert aus den im Contao Backend vorgenommenen Konfigurationen eine Definition mit der LeafletPHP Bibliothek. Dies ist im Grunde eine Abstraktion dessen, wie man die Karte mit der originalen Javascript-Bibliothek definieren würde. Diese Definition wird letztendlich zu Javascript und GeoJSON Daten konvergiert.

Dieser Aufbau ermöglicht eine flexible Manipulation sämtlicher Definitionen, ist aber auch recht rechenintensiv. Daher unterstützt Leaflet für Contao ein Caching. Dies wird auf der Ebene der Karten definiert. Da manche Layer nicht gecacht werden sollen, z.B. da sie auf die Grenzen der Karte reagieren sollen, kann man Layer vom Caching ausschließen.

1.1.4 Karten & Layer verwalten

Leaflet für Contao trennt die Datenverwaltung von der Darstellung. Sämtliche Kartenelemente werden unter Layer verwaltet. Ob Tile-Layer, Marker, Vektoren, Markercluster, ... - alles wird unter Layer angelegt.

Mit den Karten werden die verschiedenen Layer miteinander verbunden.

Karten verwalten

Layer verwalten

In Leaflet für Contao werden sämtliche Kartenelemente in Layern organisiert.

Datei-Layer

Formate

Leaflet für Contao unterstützt neben der Definition der Karten innerhalb von Contao auch externe Daten einzubetten. Dafür werden folgende Datenformate unterstützt:

- GPS Exchange Format (GPX)
- Keyhole Markup Language (KML)
- Wellknown Text (WKT)
- GeoJSON
- TopoJSON

Die Daten werden mit Hilfe des Plugins Omnivore zu GeoJSON Features konvertiert. Unterstützt werden dabei alle Funktionen der Formate, welche die Bibliothek mit sich bringt.

Datei-Layer im Backend definieren

Exterene Dateien werden als Layer im Backend definiert. Externe Dateien werden grundsätzlich dynamisch mittels Ajax-Requests nachgeladen.

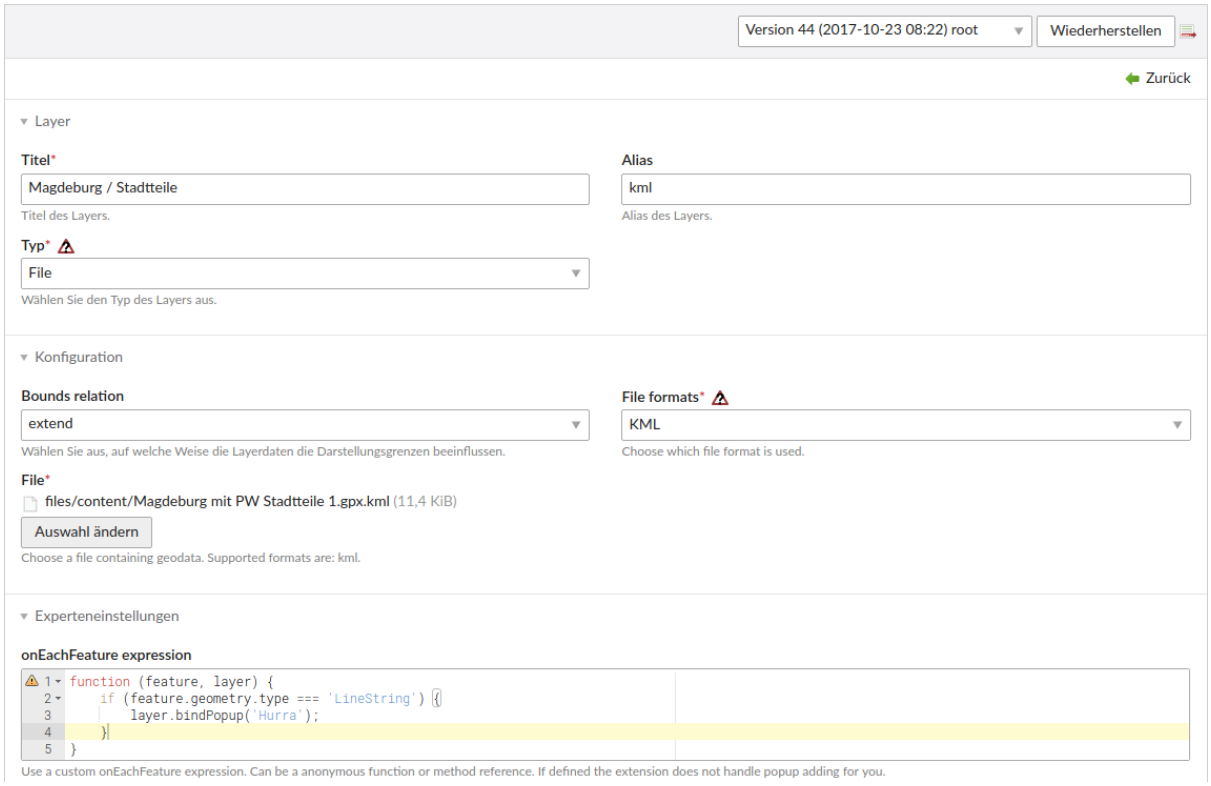

Abb. 1: Datei-Layer im Backend definieren

Mit folgenden Schritten richten Sie ein Datei-Layer ein:

- Unter Leaflet-Layer einen neuen Layer vom Typ „Datei“ (file) anlegen.
- Entsprechendes Dateiformat auswählen
- Datei auswählen
- Neues Layer den gewünschten Karten zuordnen.

neue Dateien hochgeladen werden sollen.

Kartengrenzen

Als Einflussfaktor auf die Kartengrenzen (Bounds mode) steht die Option *extend* (Bounds relation) zur Verfügung. Die Kartengrenzen werden demnach abhängig von den Daten erweitert.

Callbacks

Wie gewohnt können mithilfe der pointToLayer und onEachFeature Callbacks die Darstellung der Daten beeinflusst werden.

1.1.5 Karten auf der Website einbinden

1.1.6 Anpassen

Leaflet für Contao ist so konzipiert, dass es für

Mit den Karten werden die verschiedenen Layer miteinander verbunden.

Verhalten von Markern individualisieren

Leaflet für Contao nutzt für Marker als *Datenformat GeoJSON*. Dies bedeutet, dass Marker nicht über die Javascript API von Leaflet vorgerendert übergeben werden, sondern vielmehr ein GeoJSON Feature übermittelt wird, dass dann als Marker gerendert wird.

Dieser Aufbau hat den Vorteil, dass im GeoJSON Feature beliebige Daten mitgegeben werden können und damit die Darstellung individualisiert werden kann.

Mit dem Einstellungsmöglichkeiten *pointToLayer expression* im Kartenlayer und *featureData* im Marker selber kann direkt über das Backend eine Anpassung erfolgen.

Beispiel: Marker mit Link versehen

Besteht die Anforderung einen Marker direkt zu verlinken, kann wie folgt vorgegangen werden:

1. Link hinterlegen

Zusätzliche Daten können als *JSON* im Eingabefeld *featureData* hinterlegt werden. Diese werden dann im GeoJSON-Feature als Property *data* hinzugefügt.

```
{
    "href": "https://contao.rocks"
}
```

Im GeoJSON-Feature existiert nun folgende Eigenschaft *feature.properties.data.href*.

2. *pointToLayer* Expression definieren

Um einen Marker nun zu verlinken, kann im Marker-Layer der pointToLayer angepasst werden:

```
function (geoJsonPoint, latlng) {
    // Erstelle einen Marker anhand der Standard-Routine der Contao-Leaflet
↪Erweiterung
    var marker = L.contao.pointToLayer(geoJsonPoint, latlng);

    // Falls in den GeoJSON Properties ein Link gesetzt ist, füge ein Click-Event
↪hinzu
    if (geoJsonPoint.properties.data !== undefined && geoJsonPoint.properties.data.
↪href !== undefined) {
        marker.on('click', function () {
            window.location.href = geoJsonPoint.properties.data.href;
        })
    }

    return marker;
}
```

1.1.7 Erweiterungen

MetaModels für Leaflet Maps

Mit der Leaflet-Maps Integration wird die Darstellung von MetaModels in der Erweiterung netzmacht/contao-leaflet-maps ermöglicht.

Bemerkung: Diese Dokumentation bezieht sich ausschließlich auf Contao 4, auch wenn die Erweiterung auch für Contao 3.5. bereitgestellt wird.

Funktionen

- MetaModels Item als Marker auf Karte rendern
- Im MetaModels Item Layer referenzieren und auf der Karte darstellen
- Im MetaModels Item GeoJson-Dateien verlinken und auf der Karte darstellen
- Attribut Leaflet-Karte: Direkt eine Karte im MetaModels Item rendern

Voraussetzungen

Contao 4

- min. Contao 4.4
- MetaModels 2.1
- netzmacht/contao-leaflet-maps 3.0
- min. PHP 7.1
- min. Symfony 3.4

Contao 3.5

(Bugfixsupport auslaufend im Mai 2019)

- MetaModels 2.0

- netzmacht/contao-leaflet-maps 2.0
- min. PHP 5.4

Installation

Über Composer/Contao Manager lässt sich netzmacht/contao-leaflet-metamodels in der Version **~3.0.0-beta1** *(Stand 08.02.2019)* installieren.

MetaModel auf Karte integrieren

In dieser Anleitung wird gezeigt, wie man ein MetaModels, welches Geokoordinaten besitzt, auf einer Karte von Leaflet für Contao dargestellt werden kann.

Koordinaten-Attribute

Die Geokoordinaten können als getrennte Attribute oder in einem Attribut (Latitude und Longitude mit Komma getrennt) im MetaModel definiert werden. Als Attributstyp eignet sich z.B. ein einfaches Textattribut.

Abb. 2: Attribute Latitude und Longitude im MetaModel

MetaModels Layer anlegen

Als nächster Schritt, wird unter Karten-Layer einen neuen Layer vom Typ „MetaModels“ angelegt. Folgende Einstellungen sind hier vorzunehmen:

- **Typ**: MetaModel auswählen
- **MetaModel**: Das gewünschte MetaModel
- **Bounds relation**: Legt fest, welche Abhängigkeiten zwischen den Elementen des Layers und den Kartengrenzen bestehen soll - Auswahl von *extend*. Die Kartengrenzen werden durch die definierten Marker erweitert.
- **Anzuwendende Filtereinstellung**: Hier wird, wie bei MetaModels gewohnt, eine Filtereinstellung ausgewählt, die die anzuzeigenden Items beeinflusst.

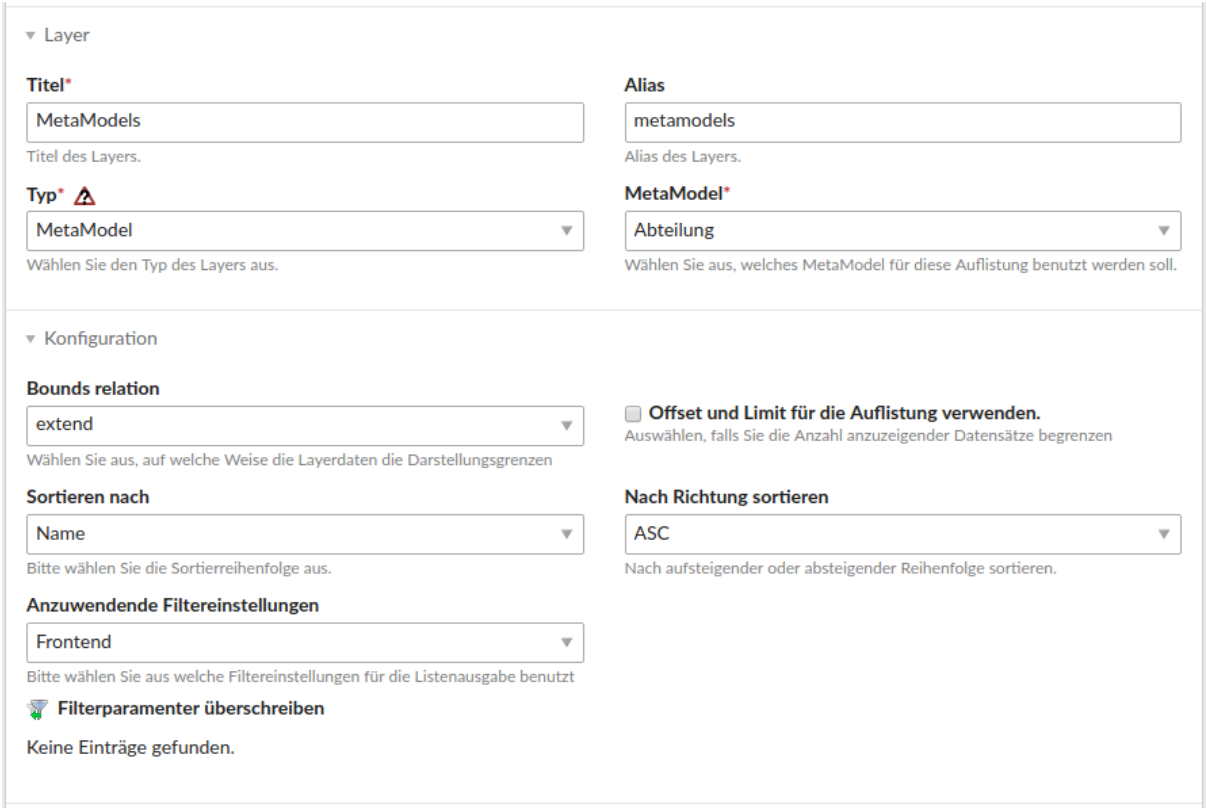

Abb. 3: Konfiguration des Layers MetaModels

MetaModels Layer Renderer anlegen

Im nächsten Schritt wird definiert, wie das MetaModels Item auf der Karte dargestellt werden soll. Diese sollen in dem Beispiel als Marker dargestellt werden. Dazu können über das Bearbeiten-Icon des Karten-Layers die entsprechenden *Renderer* angelegt werden.

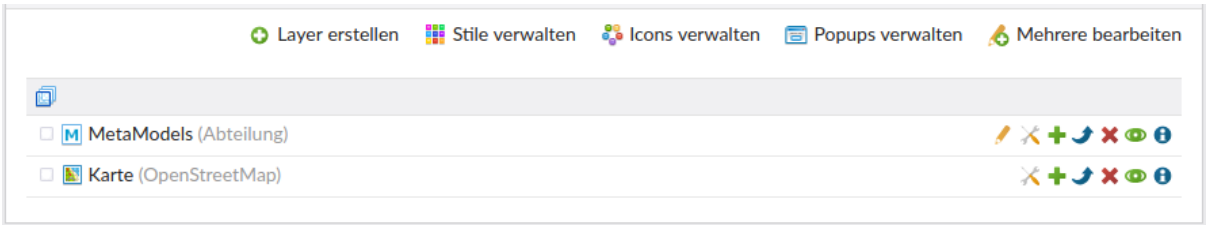

Abb. 4: Übersicht der Karten-Layer

In der Eingabemaske ist es möglich, neue Renderer zu definieren. Folgende Einstellungen sind hier vorzunehmen:

- **Typ**: Auswahl von *marker*, da die MetaModel Items als Marker dargestellt werden sollen
- **Koordinaten**: Auswahl von *separate*, wenn die Werte für Latitude und Longitude in separaten Attributen vorliegen
- **Breite-Attribut**: Auswahl des Attributs für *Latitude* aus
- **Länge-Attribut**: Auswahl des Attributs für *Longitude* aus
- **Rendererinstellung aktivieren**: aktivieren der Rendereinstellung
- **Verzögertes Laden**: Bei größeren Listen empfiehlt sich das dynamische Nachladen der Kartendaten über eine API. Diese werden dann nicht direkt als Javascript gerendert.

Zusätzlich zu der Grundkonfiguration, kann das MetaModel auch als Popup zum Marker hinzugefügt werden. Hier werden zwei Modi unterstützt:

- **render**: Eine Rendereinstellung wird ausgewählt und gerendert
- **attribute**: Es wird ein Attribut gerendert. Auch hierfür muss eine Rendererinstellung ausgewählt werden

Weiterhin ist es möglich die Darstellung als Icon zu beeinflussen. Es kann eines der vordefinierten Icons ausgewählt oder Alternativ dazu über ein MetaModels-Attribut bestimmt werden.

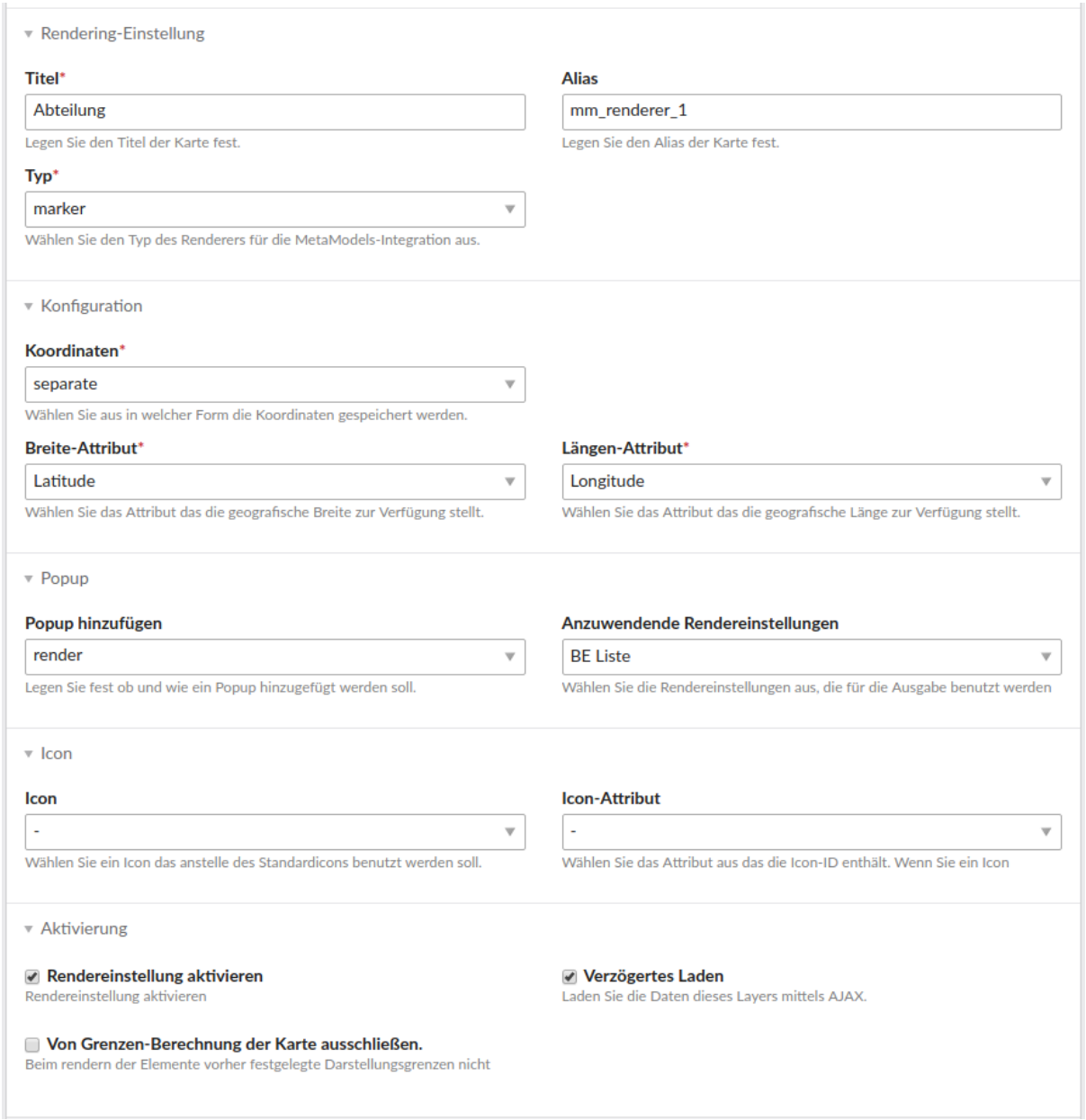

Abb. 5: Einstellung des Renderers

Layer in Karte aktivieren

Als letzter Schritt, muss dem Layer für die Darstellung noch eine Karte zugewiesen werden. Dies kann über die Standardlayer einer Karte erfolgen.

Zudem ist es zu empfehlen, bei der Funktion *Grenzen festlegen* die Optionen *bei Karteninitialisierung* und *Nach dem Laden des verzögerten Features* zu aktivieren. Damit erweitert sich die Karte dynamisch um den Bereich,

indem die Marker existieren.

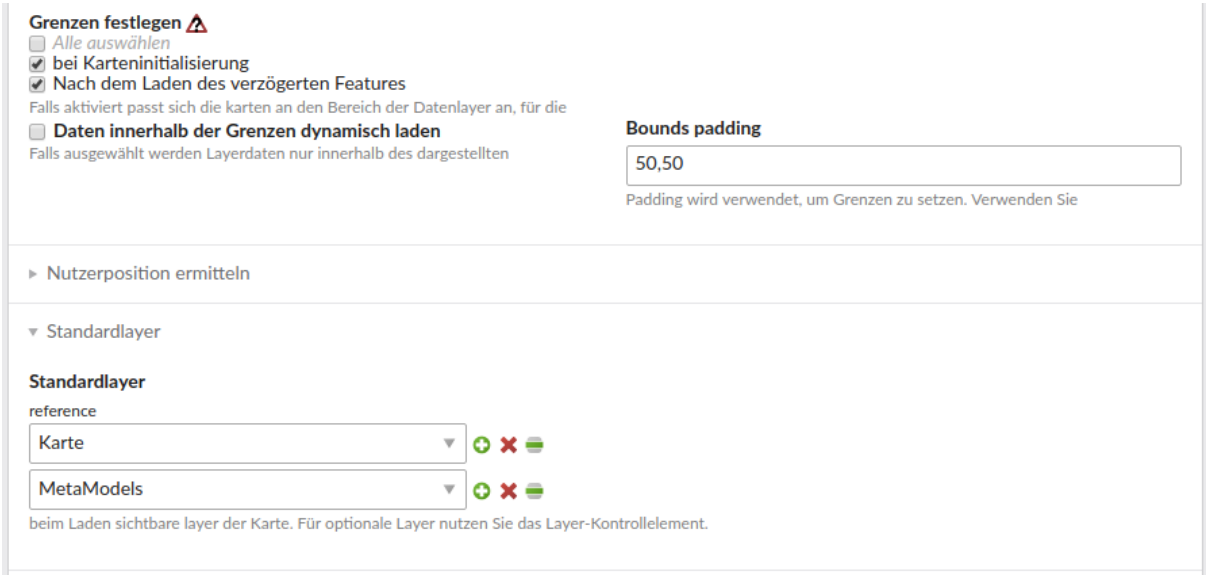

Abb. 6: Karteneinstellungen

Ist auf der Seite ein Filter eingebunden der die oben ausgewählte Filtereinstellung bedient, wird die Kartenansicht entsprechend gefiltert.

KAPITEL 2

Entwicklerhandbuch

2.1 Entwicklerhandbuch

KAPITEL 3

Lizenz

KAPITEL 4

Mitwirkende

KAPITEL 5

Verzeichnisse

- genindex